IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of:	Peter Michael Gits, et al.		
Confirmation No.:	9267		
Serial No.	09/882,221	Examiner:	Thomas Duong
Filed:	June 15, 2001	Group Art Unit:	2145
For:	NET LURKERS		
Date:	May 22, 2007		

Mail Stop Appeal Brief – Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

AMENDED APPELLANTS' BRIEF UNDER 37 CFR §41.37

Appeal is taken from the Examiner's Office Action mailed January 9, 2006, finally rejecting claims 11-14, 16, 18-19, 27-31, 38-45 in the instant application, upon direction of the panel response in a Pre-Appeal Brief Review, mailed July 19, 2006.

This Appeal Brief is in furtherance of the Notice of Appeal mailed in this case on May 15, 2006, with a request for Pre-Appeal Brief Request for Review that resulted in the panel response noted above.

The fees required under §41.37(a)(2) and any required petition for extension of time for filing this Brief and fees therefor are dealt with in the accompanying TRANSMITTAL OF APPEAL BRIEF.

REAL PARTY IN INTEREST UNDER 37 CFR §41.37(c)(1)(i)

The real parties in interest are Peter M. Gits, Dale J. Seavey, Hoan T. Dang and David R. Oran.

RELATED APPEALS AND INTERFERENCES 37 CFR §41.37(c)(1)(ii)

The Board's decision in the present Appeal will not directly affect, or be directly affected, or have any bearing on any other appeals or interferences known to the appellant, or to the appellant's legal representative, or the assignee.

STATUS OF CLAIMS UNDER 37 CFR §41.37(c)(1)(iii)

Status of All the Claims:

1. Claims presented: 1-45
2. Claims withdrawn from consideration but not eancelled: NONE
3. Claims eanceled: 1-10, 15, 17, 20-26, 32-37
4. Claims pending: 11-14, 16, 18-19, 27-31, 38-45

of whieh:

a. claims allowed: NONE

b. claims rejeeted: 11-14, 16, 18-19, 27-31, 38-45

All the rejeeted elaims are being appealed. The appealed claims are eligible for appeal, having been finally rejeeted.

STATUS OF AMENDMENTS UNDER 37 CFR §41.37(c)(1)(iv)

Subsequent to the last Office Aetion mailed on January 9, 2006, which contained a Final rejection of the appealed elaims, an amendment has been filed, mailed on March 9, 2006, and was entered for purposes of appeal, as noted in the Advisory Action mailed April 14, 2006.

SUMMARY OF THE CLAIMED SUBJECT MATTER UNDER 37 CFR §41.37(c)(1)(v)

Claims 11, 14, 31, 42 and 44 are the independent claims. The invention as claimed in claim 11 is a network lurking agent comprising an inquirer and a lurker, as described in the specification on page 6, lines 1-6 and Figure 3. Figure 3 shows software used by a network lurking agent. In Figure 3, network lurking agent 305 is operating on computer system 105 from Figure 1. Network lurking agent 305 includes lurker software 310 to lurk to a colleague's office (or other location, depending on the setting for Net Lurkers), inquirer software 315 to inquire as to the availability of the colleague, and receiver software 320 to receive a message from a network receiving agent.

The invention as claimed in claim 14 is a system comprising a JavaSpace persistent store, an environment setting, a network receiving agent and a network lurking agent. The network lurking agent is described above with regard to claim 11. The environment setting, referring to page 7, line 26, and Figure 4, stores the information received from devices in office 410 in environment setting 220. Environment setting 220 is stored in Space 405 within the Scalable Interface system. As network receiving agent 205 receives new sensor input from office 410, network receiving agent 205 can update environment setting 220. The network receiving agent, referring to page 5, line 22, and Figure 2, includes sensor software 210 to sense changes in an environment and updater software 215 to update environment setting 220. The operation of sensor software 210 and updater software 215 is discussed more with reference to FIG. 4 below. Network receiving agent 205 also includes receiver software 225 to receive an inquiry from a network lurking agent, history store 230 to track the history of who has attempted to contact network receiving agent 205, and message store 235 to store messages left for the user with

network receiving agent 205. A JavaSpace persistent store is described on page 4, line 4, as being an example of a persistent store with the capabilities of having objects inserted into the store such that they do not lose their attributes and of providing a notification service as the objects are inserted. JavaSpaces™ is used as the persistent stores.

Claim 31 is a computer readable medium to implement the methods of claim 42, and claim 44 is a variation on claim 42. The invention as claimed in claim 42 is a method of using a network lurking agent, comprising identifying an environment of interest, and placing an inquiry as to the availability of a user in a JavaSpace persistent store. Identifying an environment of interest is set forth in the specification on page 8, line 16, with reference to Figure 5. In FIG. 5, a user in office 505 uses network lurking agent 305 to attempt to contact a colleague in office 410. Recall that in the preferred embodiment, the lurker software includes a web browser, and an office or other location is represented as a URL. The user in office 505 lurks by entering a URL for an office the user wishes to visit. For example, to visit John Doe's office, the user might enter <http://www.eompany.com/JDoe.office> as the URL. Placing an inquiry is disclosed on page 5, line 21, with reference to Figure 5. Network lurking agent 305 places inquiry 510 in Spaee 405 to inquire as to the availability of the user in office 410. Space 405 notifies network receiving agent 205 about inquiry 510, which then takes inquiry 510 from Space 405. Network receiving agent 205 checks environment setting 220 to see if the user is in office 410.

GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL UNDER 37 CFR §41.37(c)(1)(vi)

The issue on appeal is whether claims 11-14, 16, 18-19, 27-31, and 38-45 are unpatentable under 35 U.S.C. §103(a) Saulpaugh, et al., US Patent No. 6,792,466 in view of Thiemer, et al., US Patent No. 5,493,692.

ARGUMENT UNDER 37 CFR §41.37(c)(1)(vii)

In the office action mailed January 9, 2006, it states, "Saulpaugh discloses, *an inquirer designed to place an inquiry in a JavaSpace persistent store space called a Space, the Space part of the Scalable Infrastructure system; and* (Saulpaugh, co1.2, lines 55-65; co1.4, lines 32-43; co1.6, lines 23-35; co1.8, line 55 - co1. 9, line 3; co1. 12, lines 30-48; col. 15, lines 15-29; co1.15, line 67 - col. 16, line 4; co1.16, lines 22-43; co1.42, lines 42-56; co1.43, lines 3-12; co1.46, line 53 -co1.47, line 9; col. 11, lines 37-59)..."

However, Saulpaugh specifically teaches away from JavaSpaces. See Saulpaugh, col. 3, line 29 through col. 4, line 44; col. 6, lines 23-35; and col. 85, lines 30-36, as examples. These are only some portions of Saulpaugh that addresses shortcomings in Jini and JavaSpaces, as the discussion of the shortcomings occurs throughout the text of the patent.

In response to this argument, the Examiner has made three arguments to support his position. First, the Examiner asserts that "Saulpaugh teaches of '*a space facility [that is] provided to which a client may register or unregister to obtain notification when something is added to or removed from the space*'" (Office Action dated January 9, 2006, page 16, lines 5-7, citing to Saulpaugh, column 42, lines 42-44; emphasis in original). This point is repeated where the Examiner recites that "Saulpaugh teaches of '*storage (both transient and persistent) providers are examples of services that enable clients and services to store, advertise, and address content*'" (Office Action dated January 9, 2006, page 17, lines 16-18, citing Saulpaugh, column 16, lines 24-27; emphasis in original). The Examiner goes on to argue that Saulpaugh is "improving on a known technology of utilizing JavaSpace as a persistent store for messaging" (Office Action dated January 9, 2006, page 16, lines 19-20). The Examiner also argues that "'*the*

API layer may also provide an interface for messages to communicate between objects or pass objects, such as Java objects. API's may be provided to discover an object repository or "space", find a particular object, claim and release an object, and write or take an object to or from the object repository'" (Office Action dated January 9, 2006, page 17, lines 11-15, citing Saulpaugh, column 11, lines 49-55; emphasis in original).

Second, the Examiner argues that the advances in technology enable storing large amounts of data in small devices (Office Action dated January 9, 2006, page 16, lines 16-18).

Third, the Examiner argues that Saulpaugh describes implementing the "'*message capable network layer . . . from the network classes provided by the Java2 Micro Edition (J2ME) platform . . . for smaller footprint devices that do not have the resources for a full Java platform*'" (Office Action dated January 9, 2006, page 17, lines 3-6, citing Saulpaugh, column 12, lines 30-35; emphasis in original). The Examiner continues by explaining that "'*[t]he distributed computing environment protocol definition does not require nor imply the use of Java on a device. Nor does it preclude it*'" (Office Action dated January 9, 2006, page 17, lines 8-10, citing Saulpaugh, column 15 line 57 through column 16, line 4).

There are three problems with the Examiner's response to the Applicant's arguments. First, because Saulpaugh makes it clear that Jini and JavaSpaces are inadequate to his task, Saulpaugh teaches away from using Jini and JavaSpaces. Second, the Examiner is arguing from hindsight. And third, the Examiner is citing to sections of Saulpaugh that have nothing to do with the Applicant's arguments.

The claims describe using Jini and JavaSpaces

Whatever Saulpaugh does or does not teach, Saulpaugh makes it abundantly clear that he considers Jini and JavaSpaces inadequate for the purpose of his patent. This is made clear by his

repeated remarks about the limitations of Jini and JavaSpaces. Examples of where Saulpaugh makes such remarks have been cited above. The Appellant points out that these citations are merely exemplary: Saulpaugh describes at length at other points about how Jini and JavaSpace do not provide the functionality he requires.

Thus, whatever "space facility" Saulpaugh describes, it is definitely not implemented with Jini and/or JavaSpaces. But the claims explicitly recite "a JavaSpace persistent store" (e.g., claims 11, 14, 42, and 44). In contrast, Saulpaugh expounds at length about the limitations of Jini and JavaSpaces, and makes it clear that in his opinion, his "space facility" cannot be implemented using Jini and/or JavaSpaces.

The Examiner's citations to Saulpaugh, column 11, 16, and 42 cited above, along with other citations to columns 8 and 32, if they mention a "space facility" at all, only describe "space facilities" in the abstract. Accordingly, these sections of Saulpaugh need to be read in context. Given how Saulpaugh makes it clear that, in his opinion, Jini and JavaSpaces are inadequate to the task of implementing a "space facility", the proper interpretation of these sections of Saulpaugh requires reading in Saulpaugh's understanding that the "space facility" cannot be implemented using Jini and/or JavaSpaces. This means that Saulpaugh teaches away from the claimed invention.

The Examiner is reasoning from hindsight

In responding to the Applicant's arguments, the Examiner argued that technological advances have made storage in small devices available. But at the time that Saulpaugh filed his patent application, small devices did not offer sufficient storage to support Java. This is made clear by the Examiner's citation to Saulpaugh, column 4. But in that section, Saulpaugh was, again, indicating why Jini was incapable of providing the functionality he needed, showing again

that Saulpaugh teaches away from Jini and/or JavaSpaces. But more than that, the Examiner is making a bald statement that devices today provide sufficient storage. Aside from the fact that the Examiner's statement is unsubstantiated, that devices today might provide sufficient storage does not support an obviousness rejection. In arguing that devices today are capable of providing the storage needed, the Examiner is arguing from hindsight, which is impermissible. In rejecting claims as obvious under 35 U.S.C. § 103(a), the Examiner is not permitted to argue based on hindsight. Since Saulpaugh makes it clear that devices at the time did not provide sufficient space to suit his needs, to argue that advances in technology made small devices with sufficient storage possible is arguing from hindsight.

In addition, the Examiner has completely overlooked a central word in his citation to Saulpaugh: that the devices require a certain amount of "processing". In other words, storage space alone is not enough for small devices to provide the functionality Saulpaugh describes. The Examiner has not made any comment about whether it was obvious that devices would have sufficient processing capability.

The Examiner cites to sections of Saulpaugh that are off point

In citing to columns 8, 11, 12, and 16 of Saulpaugh, the Examiner points to a number of different topics: a gate factory, a message capable network layer, and APIs. The Applicant fails to understand the Examiner's reasoning in citing these sections. None of these citations have anything to do with how the "space facility" might be constructed. Instead, these citations discuss other aspects of Saulpaugh. Because the citations have nothing to do with the construction of the "space facility" of Saulpaugh, these sections are without any merit.

In particular, the Examiner cites to the end of column 15 and the start of column 16, where Saulpaugh describes the distributed computing environment protocol definition. The

Applicant thinks it worth noting that Saulpaugh states that the definition does not require, imply, or preclude the possibility of Java on the device. The Applicant argues that this shows that Saulpaugh's "space facility", whatever it is, is not implemented in Java. If the device discussed in columns 15-16 of Saulpaugh can operate without implementing Java, then the "space facility" is not implemented using Jini and/or JavaSpaces. And if the device under can support Java or not, then that choicc is for whatever other purpose the device might want to support Java: it has nothing to do with the implementation of the "space facility".

In the Advisory Action mailed April 14, 2006, thc Examiner responded to these arguments as follows:

> *"However, the Examiner finds that the Applicants' arguments are not persuasive because Saulpaugh discloses, "the generated code may conform to the client's code execution environment (e.g. Java, C++, Smalltalk), as well as its management and security framework (Web-server and/or operating system)" (Saulpaugh, col.8, lines 8-11). This shows Saulpaugh's anticipation of utilizing Java spaces in the distributed computing environment. Again, Saulpaugh discloses his intention by stating that "in one embodiment, the gate factory may be platform independent code (e.g. Java code) executable within a virtual machine in the runtime environment of the device, and the constructed gate may comprises platform independent code executable within said virtual machine in said runtime environment of said device (Saulpaugh, col.8, lines 59-66). Furthermore, Saulpaugh discloses his anticipation of utilizing the Java spaces in stating "in one embodiment, message capable layer 106 may be implemented from the networking classes provided by the Java2 Micro Edition (J2ME) platform"(Saulpaugh, col.12, lines 30-32). In addition, Saulpaugh's figures 7, 10a, 34, and 35a-b show embodiments that utilize Java Virtual Machine and Java objects in the distributed computing environment. In particular, Saulpaugh discloses, for example, device 120 is Java based, whereas device 124 provides a native code environment" (Saulpaugh, col.14, lines 65-67). Hence, Saulpaugh's distributed computing environment is Java enabled based on the arguments and supports presented above".*

Appellants have not stated that Saulpaugh teaches away from the use of Java. Appellants' arguments stem from the requirement in the claims for a JavaSpace, which has been repeatedly discussed in Saulpaugh as being inadequate to the tasks described therein. The specifie rcquirements of each claim are set forth below.

It should be noted that the claims are argued out of numerical order. Due to amendments made during the course of prosecution, claim 27 was amended to depend from claim 42, and claim 28 was amended to depend from claim 44. In order to simplify the dependencies, the claims are taken in order of dependency, not in numerical order, but all claims are argued separately.

Claim 11.

Claim 1 requires *"an inquirer designed to place an inquiry into a JavaSpace persistent store...* and *a lurker designed to retrieve from the JavaSpace persistent store...*

As discussed at length above, Saulpaugh does not teach JavaSpaces and actually teaches away from JavaSpaces. Thiemer does not address spaces or any analogous element at all. Therefore, the combination of references does not teach, show or suggest the requirements of claim 1.

Claim 12.

Claim 12, depending from claim 11, also requires a JavaSpace persistent store, and further requires that the lurker have *a sender designed to send a message*, essentially to operate on a response retrieved from the JavaSpace. As discussed above, the combination of references does not show, teach or suggest use of a JavaSpace and actually teaches away from JavaSpaces, and therefore does not show, teach or suggest the requirements of claim 12.

Claim 13.

Claim 13, depending from claim 11, also requires a JavaSpace persistent store, and further requires that the lurker have a received *to receive a message from a JavaSpace.* As discussed above, the combination of references does not show, teach or suggest use of a JavaSpace and actually teaches away from JavaSpaces, and therefore does not show, teach or

suggest the requirements of claim 13.

Claim 14.

Claim 14 claims a system in which one element in the system is *a JavaSpace persistent store*. As discussed above, the combination of references teaches away from the use of JavaSpaces. The combination of references therefore does not show, teach or suggest the requirements of claim 14.

Claim 16.

Claim 16 depends from claim 14, and therefore also requires a *JavaSpace persistent store*. Further, claim 16 requires that the system *notifies the network receiving agent about the inquiry when the network lurking agent places the inquiry in the JavaSpace persistent store*. As discussed above, the combination of references teaches away from the use of JavaSpaces. The combination of references therefore does not show, teach or suggest the requirements of claim 16.

Claim 18.

Claim 18 depends from claim 14, and therefore requires a JavaSpace. Further claim 18 requires that *the network receiving agent and the network lurking agent are designed to open devices as a result of the inquiry, the devices enabling communication*, where the inquiry was placed in the JavaSpace in claim 14. As discussed above, the combination of references teaches away from the use of JavaSpaces. The combination of references therefore does not show, teach or suggest the requirements of elaim 18.

Claim 19.

Claim 19 depends from claim 14 and therefore requires a JavaSpace. Further claim 19 requires that *the network lurking agent is designed to place a message in the JavaSpace*. As

discussed above, the combination of references teaches away from the use of JavaSpaces. The combination of references therefore does not show, teach or suggest the requirements of claim 19.

Claim 42.

Claim 42 sets out a method for using a network lurking agent to electronieally lurk to an environment in a system, wherein the method includes *placing an inquiry as to the availability of a user in the environment of interest in a JavaSpace persistent store, the JavaSpace persistent store part of the system.* Therefore, claim 42 requires a JavaSpace. As discussed above, the combination of references teaches away from the use of JavaSpaces. The combination of references therefore does not show, teach or suggest the requirements of claim 42.

Claim 27.

Claim 27 depends from claim 42 and therefore requires the use of a JavaSpace. Further, claim 27 requires that the method further comprises *responding to the inquiry by a network receiving agent*, where the inquiry was placed into a JavaSpace in claim 42. As discussed above, the combination of references teaches away from the use of JavaSpaces. The combination of references therefore does not show, teach or suggest the requirements of claim 27.

Claim 28.

Claim 28 depends from claim 27, and therefore requires the use of a JavaSpace. Claim 28 further includes *responding to the inquiry includes accessing devices by the network lurking agent and the network receiving agent to enable communication* where the inquiry was an inquiry placed into a JavaSpace in claim 42, from which claim 27 depends. As diseussed above, the combination of references teaches away from the use of JavaSpaces. The combination of references therefore does not show, teach or suggest the requirements of claim 28.

Claim 29.

Claim 29 depends from claim 27, and therefore requires the use of a JavaSpace. Claim 29 further includes *placing a message in the JavaSpace persistent store by the network lurking agent* and *retrieving the message from the JavaSpace persistent store by the network receiving agent*... As discussed above, the combination of references teaches away from the use of JavaSpaces. The combination of references therefore does not show, teach or suggest the requirements of claim 29.

Claim 30.

Claim 30 depends from claim 27, and therefore requires the use of a JavaSpace. Claim 30 further includes *placing a message in the JavaSpace persistent store by the network receiving agent,* and *retrieving the message from the JavaSpace persistent store by the network lurking agent.* As discussed above, the combination of references teaches away from the use of JavaSpaces. The combination of referenees therefore does not show, teach or suggest the requirements of claim 30.

Claim 43.

Claim 43 depends from claim 27, and therefore requires the use of a JavaSpace. Claim 43 further requires *responding to the inquiry includes determining the availability of a user in the environment according to an environment setting in the JavaSpace persistent store.* As discussed above, the combination of references teaches away from the use of JavaSpaces. The eombination of referenees therefore does not show, teach or suggest the requirements of claim 30.

Claim 31.

Claim 31 requires *a computer-readable medium containing a program to use a network*

lurking agent to electronically lurk to a location on a computer system, the program being executable on the computer system to implement the method of claim 42. The combination of references does not show, teach or suggest the requirements of claim 31 for the reasons as applied to claim 42.

Claim 44.

Claim 44 requires *an apparatus for using a network lurking agent to electronically lurk to an environment in a system,* including *means for placing an inquiry as to the availability of a user in the environment of interest in a JavaSpace persistent store, the JavaSpace persistent store part of the system.* As discussed above, the combination of references teaches away from the use of JavaSpaces. The combination of references therefore does not show, teach or suggest the requirements of claim 44.

Claim 38.

Claim 38 depends from claim 44, and therefore requires the use of a JavaSpace. Claim 38 further requires *means for responding to the inquiry by a network receiving agent.* The inquiry is placed in a JavaSpace persistent store from claim 44. As discussed above, the combination of references teaches away from the use of JavaSpaces. The combination of references therefore does not show, teach or suggest the requirements of claim 38.

Claim 39.

Claim 39 depends from claim 38, and therefore requires the use of a JavaSpace. Claim 39 further requires *the means for responding includes means for accessing devices by the network lurking agent and the network receiving agent to enable communication.* The means for responding is for responding to an inquiry placed in a JavaSpace. As discussed above, the combination of references teaches away from the use of JavaSpaces. The combination of

references therefore does not show, teach or suggest the requirements of claim 39.

Claim 40.

Claim 40 depends from claim 38, and therefore requires the use of a JavaSpace. Claim 40 further requires *means for placing a message in the JavaSpace persistent store by the network lurking agent* and *means for retrieving the message from the JavaSpace persistent store by the network receiving agent*. As discussed above, the combination of references teaches away from the use of JavaSpaces. The combination of references therefore does not show, teach or suggest the requirements of claim 40.

Claim 41.

Claim 41 depends from claim 38, and therefore requires the use of a JavaSpace. Claim 41 further requires *means for placing a message in the JavaSpace persistent store by the network receiving agent* and *means for retrieving the message from the JavaSpace persistent store by the network lurking agent*. As discussed above, the combination of references teaches away from the use of JavaSpaces. The combination of references therefore does not show, teach or suggest the requirements of claim 41.

Claim 45.

Claim 45 depends from claim 38, and therefore requires the use of a JavaSpace. Claim 45 further requires *the means for responding includes means for determining the availability of a user in the environment according to an environment setting in the JavaSpace persistent store*. As discussed above, the combination of references teaches away from the use of JavaSpaces. The combination of references therefore does not show, teach or suggest the requirements of claim 45.

CLAIMS APPENDIX UNDER 37 CFR §41.37(c)(1)(viii)

The text of the claims on appeal is presented below.

11. A network lurking agent operable in a system, the network lurking agent comprising:

an inquirer designed to place an inquiry in a JavaSpace persistent store, the JavaSpace persistent store part of the system; and

a lurker designed to retrieve from the JavaSpace persistent store a response to the inquiry to determine the availability of a user in an environment.

12. A network lurking agent according to claim 11, the network lurking agent further comprising a sender designed to send a message when the response indicates the user is not in available in the environment.

13. A network lurking agent according to claim 11, the network lurking agent further comprising a receiver designed to receive a message from the JavaSpace persistent store.

14. A system designed to support network lurking, the system comprising:

a JavaSpace persistent store;

an environment setting stored in the JavaSpace persistent store, the environment setting including the availability of a device in an environment;

a network receiving agent designed to receive an inquiry about the availability of the device in the environment from the JavaSpace persistent store; and

a network lurking agent designed to place the inquiry in the JavaSpace persistent store.

16. A system according to claim 14, wherein the system notifies the network receiving agent about the inquiry when the network lurking agent places the inquiry in the JavaSpace persistent store.

18. A system according to claim 14, wherein the network receiving agent and the network lurking agent are designed to open devices as a result of the inquiry, the devices enabling communication.

19. A system according to claim 14, wherein;

the network lurking agent is designed to place a message in the JavaSpace persistent store if the inquiry is refused; and

the network receiver is designed to refuse the inquiry and to receive the message from the JavaSpace persistent store.

27. A method according to claim 42, the method further comprising responding to the inquiry by a network receiving agent.

28. A method according to claim 27, wherein responding to the inquiry includes accessing devices by the network lurking agent and the network receiving agent to enable communication.

29. A method according to claim 27, wherein responding to the inquiry includes:

refusing the inquiry by the network receiving agent;

placing a message in the JavaSpace persistent store by the network lurking agent;

retrieving the message from the JavaSpace persistent store by the network receiving agent; and

storing the message for later access from the environment.

30. A method according to claim 27, wherein responding to the inquiry includes:

placing a message in the JavaSpace persistent store by the network receiving agent;

retrieving the message from the JavaSpace persistent store by the network lurking agent; and

receiving the message at the network lurking agent.

31. A computer-readable medium containing a program to use a network lurking agent to electronically lurk to a location on a computer system, the program being executable on the computer system to implement the method of claim 42.

38. An apparatus according to claim 44, the apparatus further comprising means for responding to the inquiry by a network receiving agent.

39. An apparatus according to claim 38, wherein the means for responding includes means for accessing devices by the network lurking agent and the network receiving agent to enable communication.

40. An apparatus according to claim 38, wherein the means for responding includes:

means for refusing the inquiry by the network receiving agent;

means for placing a message in the JavaSpace persistent store by the network lurking agent;

means for retrieving the message from the JavaSpace persistent store by the network receiving agent; and

means for storing the message for later access from the environment.

41. An apparatus according to claim 38, wherein the means for responding includes:

means for placing a message in the JavaSpace persistent store by the network receiving agent;

means for retrieving the message from the JavaSpace persistent store by the network lurking agent; and

means for receiving the message at the network lurking agent.

42. A method for using a network lurking agent to electronically lurk to an environment in a system, the method comprising:

identifying an environment of interest; and

placing an inquiry as to the availability of a user in the environment of interest in a JavaSpace persistent store, the JavaSpace persistent store part of the system.

43. A method according to claim 27, wherein responding to the inquiry includes determining the availability of a user in the environment according to an environment setting in the JavaSpace persistent store.

44. An apparatus for using a network lurking agent to electronically lurk to an environment in a system, the apparatus comprising:

means for identifying an environment of interest; and

means for placing an inquiry as to the availability of a user in the environment of interest in a JavaSpace persistent store, the JavaSpace persistent store part of the system.

45. An apparatus according to claim 38, wherein the means for responding includes means for determining the availability of a user in the environment according to an environment setting in the JavaSpace persistent store.

EVIDENCE APPENDIX UNDER 37 CFR §41.37(c)(1)(ix)

No evidence external to the record of this case has been submitted or relied upon.

RELATED PROCEEDINGS APPENDIX UNDER 37 CFR §41.37(c)(1)(x)

As stated above, there are no related proceedings.

CONCLUSION

The Appellant requests favorable consideration by the Board. If any questions remain, please call the undersigned.

Respectfully submitted,

MARGER JOHNSON & McCOLLOM, P.C.

Julie L. Reed

Julie L. Reed
Reg. No. 35,349

MARGER JOHNSON & McCOLLOM, P.C.
210 SW Morrison Street, Suite 400
Portland, OR 97204
503-222-3613